1974

A methodology for the simulation of internal control systems.

Pon, Carl Maynard.

Monterey, California. Naval Postgraduate School

A METHODOLOGY FOR THE SIMULATION OF INTERNAL CONTROL SYSTEMS

Carl Maynard Pon

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A METHODOLOGY FOR THE SIMULATION
OF INTERNAL CONTROL SYSTEMS

by

Carl Maynard Pon

June 1974

Thesis Advisor: David C. Burns

Approved for public release; distribution unlimited.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A METHODOLOGY FOR THE SIMULATION OF INTERNAL CONTROL SYSTEMS		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis June 1974
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Carl Maynard Pon		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE June 1974
		13. NUMBER OF PAGES 73
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer Simulation; Inventory Accounting Systems;

Internal Control; SIMSCRIPT II.5

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This thesis illustrates a methodology for the simulation of internal control systems. The methodology is applied to an inventory accounting system, which was the subject of the doctoral dissertation of David C. Burns, to provide a quantitative measure of the adequacy of internal control. The methodology consists of conceptualizing errors in a general manner and describing these errors in routines. A computer

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
(Page 1) S/N 0102-014-6601 |

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Block 20 - ABSTRACT (Cont.)

program is then constructed which calls these routines to simulate the error processes of the internal control system being modeled. The thesis includes a discussion of the suitability and practicability of SIMSCRIPT II.5 to the simulation of internal control systems and also discusses some issues which must be resolved before it will be possible to develop basic requirements for a simulation language specially designed for the simulation of internal control systems.

A Methodology for the Simulation of Internal Control Systems

by

Carl Maynard Pon
Lieutenant (junior grade), United States Naval Reserve
B.A., University of California, Los Angeles, 1972

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
June 1974

ABSTRACT

This thesis illustrates a methodology for the simulation of internal control systems. The methodology is applied to an inventory accounting system, which was the subject of the doctoral dissertation of David C. Burns, to provide a quantitative measure of the adequacy of internal control. The methodology consists of conceptualizing errors in a general manner and describing these errors in routines. A computer program is then constructed which calls these routines to simulate the error processes of the internal control system being modeled. The thesis includes a discussion of the suitability and practicability of SIMSCRIPT II.5 to the simulation of internal control systems and also discusses some issues which must be resolved before it will be possible to develop basic requirements for a simulation language specially designed for the simulation of internal control systems.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

A. GENERAL

1. Importance of Adequate Internal Control

The existence of adequate internal control is essential to the management of any enterprise. The importance of internal control is due to the fact that almost every type of business decision involves accounting data to some extent. Adequate internal control assures management that the accounting data used in making decisions is reasonably accurate. In addition, internal control promotes operational compliance with business decisions that are established as policy.

Adequate internal control is equally important to an external auditor. The adequacy of internal control is the most important factor in determining the nature of the auditor's examination. This statement can be supported by the fact that generally accepted auditing standards require an auditor to perform a study and evaluation of internal control before he determines the auditing procedures necessary to formulate an opinion on the fairness of financial statements.

2. Definition of Adequate Internal Control

The definition of internal control used in this thesis is the definition implicitly used by the Committee on Auditing Procedure: ". . . the plan of organization and the procedures and records that are concerned with the

safeguarding of assets and the reliability of financial records. . ."[1] The definition continues with mention that internal control should provide "reasonable assurance" that internal control will achieve its objectives. This concept recognizes that the cost of internal control should not be greater than the benefit it provides. It should be noted that the above definition of "internal control" actually describes a system of internal control. In this thesis the phrase "internal control" is synonymous with "system of internal control."

When an auditor evaluates how adequate internal control is, he concentrates on the extent to which internal control prevents or detects material errors and irregularities in the financial statements. The criteria used to determine whether or not internal control is adequate is the presence or absence of a material weakness in the system. Quoting the Committee on Auditing Procedure:

> In this context, a material weakness means a condition in which the auditor believes the prescribed procedures or the degree of compliance with them does not provide reasonable assurance that errors or irregularities in amounts that would be material in the financial statements being audited would be prevented or detected within a timely period by employees in the normal course of performing their assigned functions.[2]

---

[1] Committee on Auditing Procedure, Statement on Auditing Standards No. 1, p. 20, American Institute of Certified Public Accountants, Inc., 1973.

[2] Ibid., p. 30.

3. Problem of Assessing Adequacy

According to the Committee on Auditing Procedure, the auditor should assess the adequacy of internal control using the following four step process:

a. Consider the types of errors and irregularities that could occur.
b. Determine the accounting control procedures that should prevent or detect such errors and irregularities.
c. Determine whether the necessary procedures are prescribed and are being followed satisfactorily.
d. Evaluate any weaknesses--i.e., types of potential errors and irregularities not covered by existing control procedures--to determine their effect on (1) the nature, timing or extent of auditing procedures to be applied and (2) suggestions to be made to the client.[3]

The problem which faces the auditor is that the assessment process must consider the interaction of a multitude of people, processes, and procedures. The auditor is required to estimate acceptable error rates and error magnitudes for each person, process, and procedure connected with internal control. These acceptable error rates are used as a basis for determining upper precision limits for statistical sampling tests which are intended to determine the actual rates. To derive these acceptable error rates and magnitudes, the auditor must assess their net affect on the accounting system. The more extensive the system of internal control is, the more complex such mental assessments become.

For a large enterprise, such as a Naval Supply Center, internal control systems are so extensive that it is

---

[3] *Ibid.*, pp. 31-32.

difficult to follow even one transaction through the system. Consequently, when it is necessary to consider the numerous transactions which occur over an extended period of time, mental assessment becomes extremely difficult, especially if offsetting or compound errors are possible. Since evaluation of internal control requires the auditor to consider the entire system instead of just individual errors, it can be seen that mental assessment becomes a hopelessly complex problem.

4. Need for Objective Evaluation Methods

Because of the difficulty of evaluating an entire system of internal control, the auditor may instead choose to evaluate subsystems of internal control. An example of such a subsystem would be all controls related to an inventory account. When acceptable error rates are established for a subsystem, a two-step evaluation is normally performed. First, each control is assessed separately with regard to acceptable error rates and magnitudes. Second, the net effect of all the individual acceptable error rates upon the accounting system is subjectively determined. If this net error is too large, the auditor may have to repeat the first step.

At present, auditors do not use any type of objective method to assist them in assessing the adequacy of an internal control subsystem. Consequently, the more complex the subsystem, the less confidence the auditor can place in his subjective assessment. This fact has been established by

at least one experiment which showed that even with perfect knowledge of error rates an auditor could not make an accurate subjective assessment of internal control with respect to inventory account balances.[4] This experiment illustrated a need for quantitative or objective methods of evaluating internal control which is also expressed in a forthcoming article by John Neter and Seongjae Yu.[5]

B. PURPOSE

1. Methodology for Simulating Internal Control

The first purpose of this thesis is to illustrate the use of a modified version of the methodology used by Burns in his doctoral dissertation, with the intent of simplifying the writing of a computer program for a simulation model of internal control. The methodology used by Burns was essentially to begin from scratch: the internal control system of a hypothetical firm was first simplified and then described by a computer program which defined each error process in this particular system. Consequently, this methodology resulted in a very special-purpose model.

This thesis deals with the same system, but describes it using a different methodology. Rather than describe each

---

[4] David C. Burns, Audit Evidence Evaluation Using Computer Simulation with Special Emphasis on Ascertaining the Reliability of Accounting Data, Doctoral Dissertation, Indiana University, Bloomington, Graduate School of Business, 1971.

[5] This article will appear in a forthcoming issue of The Journal of Accounting Research, published by the University of Chicago.

individual error process as it occurs in this particular system, error processes are described in routines which will be called by a main program at the proper time. These routines are so general that they require the user to specify values for parameters which were constants in Burns' dissertation (e.g., error rates and error magnitudes). Consequently, these routines can be used to describe any number of internal control systems simply by changing the sequence of error routines or by changing the values of parameters.

2. <u>Suitability and Practicability of SIMSCRIPT</u>

The second purpose of this thesis is to determine both the suitability and the practicability of SIMSCRIPT as a programming language to be used for describing simulation models of internal control systems. The reasons for choosing SIMSCRIPT rather than any other programming language are presented later in the thesis.

3. <u>Basic Requirements for an Internal Control Simulation Language</u>

The third purpose of this thesis is to identify the basic requirements for a simulation language which could be used specifically for simulation of internal control systems. Discussion of these requirements considers the basic issues and problems which must be resolved to develop such a language.

C. METHOD

1. Model an Inventory Accounting and Control System

The model presented in this thesis is based upon the inventory accounting and control system of a hypothetical firm which was described in the doctoral dissertation of David Burns. The rationale for using this system is that it presents a variety of auditing problems which are representative of the problems found in other types of internal control subsystems. In addition, since some of the most complex auditing problems are related to manufacturing inventories, the argument can be made that if this system can be modeled, any subsystem can be modeled.

2. Conceptualize Errors in a General Manner

The error processes which will be described in routines will be conceptualized in as general a manner as is possible. This means that there will be no constants in the error routines, only variables. As a result of this conceptualization, only three error routines will be needed to describe the eight error processes found in the hypothetical system.

3. Describe Routines in SIMSCRIPT

The use of SIMSCRIPT to describe error routines is completely independent of the manner in which the error processes are conceptualized. This means that SIMSCRIPT could just as easily have been used to describe the hypothetical system as originally conceptualized and that other programming languages could have been used to describe the

conceptualization of errors in a general manner. The reason for making this point is to emphasize that the use of SIMSCRIPT was an activity separate from the conceptualization of error processes as general routines.

4. Write Main Program Calling Routines

The main program, which will simulate the inventory accounting and control system, is an activity separate from the description of error routines. This means that the system described in the main program is only one of many systems that could be described by calling the same error routines in a different order and/or with different variable values.

5. Compare New Program with Original Program

The output of the "new" program (described in SIMSCRIPT) will be compared with those of the original model (described in FORTRAN). This will establish the fact that the two programs represent the same system. After this fact is established the thesis will continue with an evaluation of SIMSCRIPT and consideration of the issues and problems raised by the "new" method of conceptualizing errors.

II. ORIGINAL SIMULATION MODEL DESCRIPTION

A. THE HYPOTHETICAL FIRM

1. General

The "original" simulation model was based on the manual inventory accounting system of a hypothetical manufacturing firm.[6] The processes attributed to the hypothetical firm and its inventory accounting system were in fact abstracted from those which occurred in a real business firm. This real firm was engaged in the business of machining and selling alloy and cast-iron pipe fittings.

Since the real firm carried a product line of over two thousand fittings, it was necessary to restrict the scope of this real firm when transforming it into the hypothetical firm. Consequently, the hypothetical firm and the simulation model were restricted to four products from this total line. These four products are referred to by number, products 1, 2, 3 and 4. Production of each of these four products was assumed to involve two manufacturing departments, Department I and Department II.

The financial accounting records of the hypothetical firm carried inventories of raw materials, work-in-process, and finished goods at predetermined standard costs. As was

---

[6] A detailed description of the original simulation model can be found in Naval Postgraduate School Report 55Bu73111A, A Computer Simulation Case for the Auditing Classroom, by David C. Burns, pp. 6-66, November 1973.

mentioned before, inventory accounting operations were performed manually. A build-up of the standard costs for all four products is given in Figure 1.

2. Internal Control Weaknesses

A study and evaluation of internal control performed by an auditor on the hypothetical firm would have detected the following weaknesses in internal control with respect to inventories:

1) Receiving and inspection personnel were lax in that they did not perform physical counts of incoming raw material shipments.

2) Access to the raw material storage area was not controlled in a prudent manner.

3) The files containing standard cost cards were not maintained in an orderly fashion.

4) Foremen did not check the accuracy of production counts stated by their operators.

5) The weigh-count operator was lax in verifying the counts stated on production orders before the goods were placed in the finished goods storage area.

3. Errors Introduced to Accounting Records

The internal control weaknesses described above were further assumed to permit the following errors to affect the hypothetical firm's inventory accounting records:

1) Shipments of raw materials received by the firm sometimes contained more units than were stated on the vendor's or shipper's invoice. The receiving and inspection

FIGURE 1

STANDARD COST BUILD-UP

	Product Number 1	Product Number 2	Product Number 3	Product Number 4
Direct Material				
Type of Material	R. M. 1	R. M. 2	R. M. 3	R. M. 4
Units Required	1	1	1	1
Spoilage/Scrap, etc.	0	0	0	0
Standard Cost of Material	\$13.5000	\$16.7000	\$ 6.5000	\$ 8.000
Total/Unit	\$13.5000	\$16.7000	\$ 6.5000	\$ 8.000
Direct Labor				
Department I				
Std dir lbr hrs/unit	.06 hr	.09 hr	.04 hr	.06 hr
Std dir lbr rate	\$ 6.20/hr	\$ 6.20/hr	\$ 6.20/hr	\$ 6.20/hr
Total std dir lbr charge	\$ .3720	\$ .5580	\$ .2480	\$ .3720
Department II				
Std dir lbr hrs/unit	.04 hr	.06 hr	.04 hr	.07 hr
Std dir lbr rate	\$ 5.60/hr	\$ 5.60/hr	\$ 5.60/hr	\$ 5.60 hr
Total std dir lbr charge	\$ .2240	\$ .3360	\$ .2240	\$ .3920
Total/Unit	\$ .5960	\$ .8940	\$ .4720	\$ .7640
Burden				
Department I				
Std dir lbr hrs/unit	.06 hr	.09 hr	.04 hr	.06 hr
Std burden rate	\$12.85/hr	\$12.85/hr	\$11.40/hr	\$11.40/hr
Total std burden charge	\$ .7710	\$ 1.1565	\$ .4560	\$ .6840
Department II				
Std dir lbr hrs/unit	.04 hr	.06 hr	.04 hr	.07 hr
Std burden rate	\$51.55/hr	\$51.55/hr	\$44.05/hr	\$44.05 hr
Total std burden charge	\$ 2.0620	\$ 3.0930	\$ 1.7620	\$ 3.0835
Total Burden/Unit	\$ 2.8330	\$ 4.2495	\$ 2.2180	\$ 3.7675
Total Unit Standard Cost	\$16.9290	\$21.8435	\$ 9.1900	\$12.5315

Abbreviations: std =standard; dir =direct; lbr =labor; hrs =hours.

department personnel were assumed to be lax in performing their duties. Consequently these understatements could remain undetected and uncorrected.

2) Lack of control over access to the raw materials storage area allowed the unauthorized and unrecorded return of excess raw materials which had been requisitioned to support inflated production counts.

3) Careless maintenance of standard cost files resulted in the application of incorrect standard costs while vouching raw material purchases, costing production orders, or transferring goods from work-in-process to finished goods.

4) The laxity of foremen allowed the random overstatements of production counts to remain undetected in both Department I and Department II.

5) The laxity of the weigh-count operator allowed most of the random overstatements of production counts to pass undetected as goods were moved to the finished goods storage area.

The error processes outlined above caused both overstatements and understatements of the various inventory accounts to be introduced into the account balances. Detailed descriptions of these error processes are presented later in this thesis.

B. THE SIMULATION MODEL

1. Components

The original computer simulation model can be described in terms of the following components:

a. Framework

The framework of the model was a FORTRAN program describing the inventory accounting system. Functions performed by the program include vouching raw material purchases and the costing of material requisitions, production reports, transfers of finished goods, and sales. The program maintained two separate sets of inventory accounting records. The "reported" account balances contain the net amount of the correct balance and all errors, while the "control" account balances contain only the correct balance.

b. Input Generators

The model included several external input generators which provide the input data that is subsequently processed by the framework. These data represent the quantity of raw materials contained in raw materials shipments and the number of units of product specified in production orders.

c. Erroneous Accounting Operations

The model used the Monte Carlo technique to simulate the erroneous accounting operations by causing errors to occur at random in accordance with predefined probability distributions. These errors had an affect only on the "reported" balance, since the "control" balances reflected only the correct portion of each transaction.

d. Parameters

The parameters of the model included beginning inventory levels, files of correct and incorrect standard costs, and quantities which limited the total volume of financial accounting activity. The volume was controlled by placing a ceiling on the amount of raw materials to be purchased and on the amount of each product to be manufactured.

2. Operating Characteristics

The original model included a set of computer statements which caused 1500 iterations of the simulation process to be performed. This large number of iterations was necessary to perform certain nonparametric goodness-of-fit statistical tests. However, one hundred iterations produced statistics which were suitable for estimating the total error.

The model used pseudo-random numbers to trigger the occurance of errors during the simulation process. A different sequence of pseudo-random numbers was used during each of the 1500 iterations of the model. Consequently, the combinations of processing errors which occurred during each iteration of the model were statistically independent.

In addition to calculating the total error present in ending inventory balances at the end of each iteration, the model was designed to plot these errors as a probability distribution and to calculate the mean and standard deviation of this distribution. The mean and standard deviation of the probability distribution were quantitative measures of

the adequacy of internal control. Both measures were needed since a mean of zero dollar error could still reflect inadequate internal control if the standard deviation were large enough to make material errors reasonably likely.

III. REVISED SIMULATION MODEL DESCRIPTION

A. THE HYPOTHETICAL FIRM

1. General

The "revised" simulation model was based on the same manual inventory accounting system of the hypothetical firm. This was done so that this thesis could concentrate on the changes in the conceptualization of the system and on the different computer language used to describe the conceptualization. However, there was one minor difference between the original simulation model and the revised simulation model which had no effect on the operating characteristics of the model, but is discussed for sake of completeness.

This difference between the two models is related to the external input generators. Ceilings were established to control the total units of raw materials to be received and the units of each product that were to be put into process and transferred to finished goods. The original model reduced the quantities of a shipment or production order which would exceed the ceiling so that the ceiling would be met but not exceeded. The revised model did not reduce the quantities, so the ceiling could be slightly exceeded. However, due to the small size of individual raw material shipments and production order quantities relative to these ceilings (see Figure 2), the effect of this change on the results of the revised model was insignificant.

FIGURE 2

DISTRIBUTIONS AND CEILINGS USED
BY INPUT DATA GENERATORS

	Raw Material 1	Raw Material 2	Raw Material 3	Raw Material 4
Units to be received during the period	40,000	34,000	34,000	32,000
Mean of normal distribution of raw material shipments	200	180	200	180
Standard deviation of normal distribution of raw material shipments	25	30	25	30

	Product 1	Product 2	Product 3	Product 4
Units to be placed into process during the period	33,600	33,000	33,000	28,900
Units to be transferred to finished goods during the period	30,280	30,964	29,777	21,719
Mean of normal distribution of production orders	150	150	150	150
Standard deviation of normal distribution of production orders	35	35	35	35

2. <u>Detailed Description of Error Processes</u>

As was explained above, there was only one difference between the hypothetical firm in the original model and that in the revised model. It is important to emphasize this fact to avoid any misinterpretation of the following descriptions, which explain in detail the error processes found in the hypothetical firm and used as a basis for both the original and the revised model.

a. Receiving and Inspection Count Error

The count error which could occur during receiving and inspection operations resulted from the failure of employees to physically count shipments received. The nature of the error was the same for all four raw materials: each shipment processed had a twenty-five per cent chance of being understated by ten per cent of the correct quantity received.

b. Raw Material Receipts Pricing Error

Weak control over the standard cost file resulted in the possibility that an incorrect standard cost for raw materials could be applied to purchase orders. This in turn resulted in erroneous vouchering of raw material receipts and incorrect entries to raw materials inventory. For each of the four raw materials, there was a ten per cent chance that this error could occur. However, the monetary value of the rate error was a function of the raw material which was being vouchered. If an error occurred:

1) Raw Material 1 was priced at the standard cost of Raw Material 3.

2) Raw Material 2 was priced at the standard cost of Raw Material 4.

3) Raw Material 3 was priced at the standard cost of Raw Material 1.

4) Raw Material 4 was priced at the standard cost of Raw Material 2.

c. Raw Material Requisitions Pricing Error

Since all raw materials were put into process in Department I, pricing errors in raw material requisitions could occur only as a result of Department I operations. These pricing errors were all assumed to be the result of misfiled standard cost cards. For each of the four products (and thus, for each of the four raw materials) there was a ten per cent chance that such an error would occur. If an error occurred:

1) Raw Material 1 used to produce Product 1 was priced at the standard cost of Raw Material 3.

2) Raw Material 2 used to produce Product 2 was priced at the standard cost of Raw Material 4.

3) Raw Material 3 used to produce Product 3 was priced at the standard cost of Raw Material 1.

4) Raw Material 4 used to produce Product 4 was priced at the standard cost of Raw Material 2.

It should be noted that since the standard usage of material for each product was the same and that each product only required one raw material, errors in pricing were the only errors possible when processing raw material requisitions.

d. Department I Production Count Error

Overstatements of production counts are assumed to occur because machine operators were paid incentive wages and because internal control weaknesses made it possible for such overstatements to remain undetected. These weaknesses are due to the laxity of foremen in verifying the counts of their subordinates and the lack of control over access to the raw materials storage area. This lack of control made it possible for machine operators to plan on overstating production counts. Their method was to requisition enough material to support their inflated production counts and later return excess material without management's knowledge. In the hypothetical firm there was assumed to be a fifteen per cent chance that a "count error" would occur. When such an error did occur, the quantity stated in the production order represented a ten per cent overstatement of the actual quantity.

e. Department II Production Count Error

When a production order reaches Department II, the Department I count was always accepted as correct, even if it was overstated. Thus it was assumed that Department II employees in the hypothetical firm cover-up for the overstatements of Department I employees. In addition, Department II employees may overstate production counts regardless of whether or not Department I overstated its count on a given production order. Thus, these overstatements were independent events. The chance that a Department II employee would overstate a production order was eight per cent, and when

this "count error" occurs, the result was a five per cent overstatement of the quantity stated in the production order.

f. Standard Direct Labor Hours Rate Error

Weak control over the standard cost file resulted in the possibility that an incorrect standard for the direct labor hours to produce one unit of product could be applied when costing a job time ticket. If an incorrect standard was used, the "rate error" affected the application of both burden and direct labor to the job time ticket. The chance that such a "rate error" would occur for any product was eight per cent. When such an error did occur:

1) The standard for Product 3 was applied to a job time ticket for Product 1.

2) The standard for Product 4 was applied to a job time ticket for Product 2.

3) The standard for Product 1 was applied to a job time ticket for Product 3.

4) The standard for Product 2 was applied to a job time ticket for Product 4.

It should be noted that this error could occur in one, two, or neither of the manufacturing departments.

g. Burden Rate Error

The occurrence of a burden rate error was independent of an error in applying the standard direct labor hour rate to a job time ticket. The chance that such an error would occur was eight per cent, and when such an error did occur:

1) The burden rate for Product 3 was applied to a job time ticket for Product 1.

2) The burden rate for Product 4 was applied to a job time ticket for Product 2.

3) The burden rate for Product 1 was applied to a job time ticket for Product 3.

4) The burden rate for Product 2 was applied to a job time ticket for Product 4.

It should be noted that this error could occur in one, two, or none of the manufacturing departments.

h. Labor Rate Error

The occurrence of a labor rate error was independent of an error in applying the standard direct labor hour rate to a job time ticket. There was assumed to be a ten per cent chance that such an error would occur in the hypothetical firm. When such an error did occur, there were two possible outcomes:

1) The labor rate for last year was applied to the job time ticket.

2) The labor rate for the wrong manufacturing department was applied to the job time ticket.

These two outcomes were assumed to be equally likely. That is, each was expected to occur fifty per cent of the time. It should be noted that this error could occur in one, two, or none of the manufacturing departments.

i. Production Order Transfer Rate Error

Each production order for each type of product was transferred to finished goods until a specified number of units had been transferred. Units of product transferred to finished goods were costed at the total standard cost for the given product. Once again, the weak control over the standard cost file interjected a potential "rate error" into the accounting system. The chance of such an error occurring was eight per cent, and when this error occurred:

1) A production order for Product 1 was costed at the standard cost of Product 3.

2) A production order for Product 2 was costed at the standard cost of Product 4.

3) A production order for Product 3 was costed at the standard cost of Product 1.

4) A production order for Product 4 was costed at the standard cost of Product 2.

The standard cost relevant to this error and the errors which have been described before it are presented in Figure 3.

3. Detailed Description of Error Correction Process

Before the product stated on each production order was transferred to the finished goods storage area, a weigh-count operator verified the count stated on the production order. However, the operator was assumed to be so lax in performing his duties that only overstatements of twenty-one or more units were detected. When overstatements were

FIGURE 3

STANDARDS FOR PRODUCTION COSTING; CORRECT AND ERRONEOUS

	Product Number 1	Product Number 2	Product Number 3	Product Number 4
Direct Material				
Correct dir material std	$13.5000	$16.7000	$ 6.5000	$ 8.0000
Erroneous dir material std	$ 6.5000	$ 8.0000	$13.5000	$16.7000
Direct Labor				
Correct std dir lbr hrs/unit in Dept I	.06	.09	.04	.06
Erroneous std dir lbr hrs/unit in Dept I	.04	.06	.06	.09
Correct std dir lbr hrs/unit in Dept II	.04	.06	.04	.07
Erroneous std dir lbr hrs/unit in Dept II	.04	.07	.04	.06
Correct std dir lbr rate in Dept I	$ 6.20	$ 6.20	$ 6.20	$ 6.20
Erroneous std dir lbr rate Number I in Dept I	$ 6.00	$ 6.00	$ 6.00	$ 6.00
Erroneous std dir lbr rate Number II in Dept I	$ 5.60	$ 5.60	$ 5.60	$ 5.60
Correct std dir lbr in Dept II	$ 5.60	$ 5.60	$ 5.60	$ 5.60
Erroneous std dir lbr rate Number I in Dept II	$ 5.40	$ 5.40	$ 5.40	$ 5.40
Erroneous std dir lbr rate Number II in Dept II	$ 6.20	$ 6.20	$ 6.20	$ 6.20
Correct std dir lbr cost/unit	$ .5960	$ .8940	$ .4720	$ .7640
Erroneous std dir lbr cost/unit	$ .4720	$ .7640	$ .5960	$ .8940

This Figure is continued on the next page.

FIGURE 3 (Continued)

STANDARDS FOR PRODUCTION COSTING; CORRECT AND ERRONEOUS

	Product Number 1	Product Number 2	Product Number 3	Product Number 4
Burden				
Correct std burden rates for Dept I; in dollars/ std dir lbr hr	$12.85	$12.85	$11.40	$11.40
Erroneous std burden rates for Dept I; in dollars/ std dir lbr hr	$11.40	$11.40	$12.85	$12.85
Correct std burden rates for Dept II; in dollars/ std dir lbr hr	$51.55	$51.55	$44.05	$44.05
Erroneous std burden rates for Dept II; in dollars/ std dir lbr hr	$44.05	$44.05	$51.55	$51.55
Correct std burden cost/ unit	$ 2.8330	$ 4.2495	$ 2.2180	$ 3.7675
Erroneous std burden cost/unit	$ 2.2180	$ 3.7675	$ 2.8330	$ 4.2495
Total Unit Standard Cost				
Correct unit std cost	$16.9290	$21.8435	$ 9.1900	$12.5315
Erroneous unit std cost	$ 9.1900	$12.5315	$16.9290	$21.8435

detected, the correct count was placed on the production order. When such a correction was noted by the accounting system, two entries were made when the transfer to finished goods was recorded. First, a debit to finished goods and credit to work-in-process was made (subject to the production order transfer rate error described above). Second, an entry was made to reverse the entries assumed to have been made earlier when the nonexistent units were recorded as work-in-process. This correction itself could cause an error because it assumed the nonexistent units originated in Department I and made the correction accordingly.

B. THE SIMULATION MODEL

1. Components

The components of the revised simulation model differed from those of the original model in the following ways:

a. Framework

The framework of the model was a SIMSCRIPT program, rather than a FORTRAN program. However, the use of a different computer language was not the most significant change in the framework, since FORTRAN could have almost as easily have performed the same functions as SIMSCRIPT. The most significant difference was the approach used to conceptualize the errors before the program was written. As has been previously mentioned, the original program was used to describe a conceptualization of one particular accounting

system in a statement by statement fashion. The revised program, on the other hand, described this particular system by calling a series of general routines which were conceptualized in a way such that they might be used to describe any number of systems. This approach was made possible by conceptualizing errors in as general a manner as was possible.

One example of the generality implicit in this new program was the fact that the first statement of the program required the user to specify the number of accounts which are to be present in the accounting system to be simulated. Accounting transactions are recorded in this system by calling one of the three ENTRY routines and specifying accounts to be debited and credited. The only difference between the three ENTRY routines are housekeeping details which could have been handled in the main program. It should be noted that the ENTRY routines record only the errors which might occur in processing a given type of accounting transaction.

This fact represents another difference between the framework of the original model and that of the revised model. In the original model, the total error was determined by subtracting the individual "control" account balances from the "reported" account balances. In the revised model, only a record of the error content of the accounts was maintained. Consequently, only one set of accounts was needed.

b. Input Generators

While the original model required that several input generators be built into the program, the revised model only used one. This was possible because both raw material shipments and the production orders were assumed to be normally distributed. Shipments of Raw Materials 1 and 3 had a mean of 200 and a standard deviation of 25, while Raw Materials 2 and 4 had a mean of 180 and a standard deviation of 35. All production orders had a mean of 150 and a standard deviation of 35. A random number from a normal distribution with a prescribed mean and standard deviation was obtained by calling the QUANT1 routine. The number determined by the routine was then stored in a memory location which could be accessed from either the main program or other routines.

c. Erroneous Accounting Operations

While the original model required separate groups of computer statements for each of the nine error processes which were previously described in detail, the revised model used only three general routines to simulate the nine erroneous accounting operations. Detailed descriptions of the three general error routines follow:

(1) <u>Routine ERROR1</u>. This routine required the specification of five arguments: the probability a count error would occur, the magnitude, in decimal form (i.e., .10 equals 10% overstatement) of such a count error, the probability a rate error would occur, the correct rate to be

used, and the incorrect rate which would be used if a rate error occurred. Besides being used when both a count error and a rate error could occur, this routine could be used where only one kind of error occurred by assigning a value of zero to nonapplicable arguments. In addition, a predetermined count error could be used by assigning a negative value to the probability of a count error.

Routine ERROR1 could simulate six of the nine error processes present in the original simulation. The only errors which this routine could not simulate were the standard direct labor hours rate error, the burden rate error, and the labor rate error. It should be mentioned that even these three error processes could have been simulated in ERROR1 by increasing the number of arguments and the complexity of the routine. The reason this was not done was simply an arbitrary decision that these three errors did not seem to fit logically into the routine.

(2) Routine ERROR2. This routine required the specification of six arguments: the probability of a rate error regarding the standard direct labor hours, the correct standard direct labor hours, the correct standard for direct labor hours per unit of product, the incorrect standard that would be used if an error occurred, the probability of a rate error regarding the standard burden rate, the correct standard burden rate, and the incorrect standard burden rate used if such an error occurred. It should be pointed out that if the standard burden rate for the hypothetical firm

had been stated in terms of cost per unit of output rather than cost per hour, ERROR1 could have been used instead of ERROR2. It should also be noted that if a combined rate for burden and labor costs were applied to a production order, only one error routine would be needed. Finally, it should be noted that the count error determined in routine ERROR1 was also reflected in the error generated in ERROR2.

In summary, the routine ERROR2 which was used in the revised model simulated two of the nine error processes which were previously described in detail. These two errors were the standard direct labor hours rate error and the burden rate error. Consequently, the only error process left to be simulated was the labor rate error.

(3) Routine ERROR3. This routine required the specification of five arguments: the probability of such a rate error, the first incorrect rate that might be used, the second incorrect rate that might be used, the probability that the first incorrect rate would be used (assuming the error did occur), and the correct rate. The probability of the second incorrect rate being used (assuming the error did occur) was equal to one minus the probability of the first incorrect rate being used. It should be pointed out that each time ERROR3 was called, it had to be preceeded by ERROR2. This was necessary since the labor rate was expressed in terms of cost per hour rather than cost per unit produced, so the standard direct labor hour rate used in ERROR2 was also needed in ERROR3. As was the case with ERROR2, any

count error generated in ERROR1 was reflected in the error generated in ERROR3. It is also worth mentioning the fact that if only one incorrect labor rate were possible, ERROR3 would not have been required, since ERROR2 or ERROR1 could then have simulated the labor rate error process.

d. Parameters

The revised model included all parameters contained in the original model with the exception of beginning inventory levels. This parameter was not needed since the revised model dealt only with errors, and the error in the beginning inventory was assumed to be zero. Due to its more general nature, the revised model explicitly included a few parameters which were only implicitly included in the original model. These parameters were the rate of occurrence of each of the nine error processes previously described and the magnitude of each of the three count errors.

2. Output Comparison

In order to provide reasonable verification that the same hypothetical firm was described in both the original and the revised simulation model, the output generated by each of the two models were compared. The output (see Figure 4) of the two programs compared as follows:

1) For the error in the ending balance of the raw material account, the original model produced a mean of -$56,449 and a standard deviation of $15,659, while the revised model produced a mean of -$59,265 and a standard deviation of $17,922.

FIGURE 4

OUTPUT OF ORIGINAL AND REVISED SIMULATION MODELS

	Arithmetic Mean of Error	Standard Deviation of Error
Raw Materials Inventory:		
Original Model	-$56,449	$15,659
Revised Model	-$59,265	$17,922
Work-in-Process Inventory:		
Original Model	$ 2,373	$15,736
Revised Model	-$ 1,185	$15,891
Finished Goods Inventory:		
Original Model	$17,060	$11,230
Revised Model	$23,207	$10,136
Total Combined Inventory:		
Original Model	-$37,040	$11,426
Revised Model	-$37,243	$13,872

2) For the error in the ending balance of the work-in-process account, the original model produced a mean of $2,373 and a standard deviation of $15,736, while the revised model produced a mean of -$1,185 and a standard deviation of $15,891.

3) For the error in the ending balance of the finished goods account, the original model produced a mean of $17,060 and a standard deviation of $11,230, while the revised model produced a mean of $23,207 and a standard deviation of $10,136.

4) For the error in the ending balance of the combined inventory account, the original model produced a mean of -$37,040 and a standard deviation of $11,426, while the revised model produced a mean of -$37,243 and a standard deviation of $13,872.

It should be noted that the output generated by the original model was the result of 1500 replications, while that of the revised model was the result of only 100 replications.

Although some of the differences between the output of the original model and that of the revised model might be statistically significant, in the aggregate they would not be considered to be material by an auditor. For example, it can be stated with ninety-nine percent confidence that there is no difference between the mean of the error in the combined inventory account of the original model and that in the revised model. The large size of the standard deviations relative to the means provide further support to the

statement that an auditor would find no material differences between the results of the original model and those of the revised model.

IV. SUITABILITY AND PRACTICABILITY OF SIMSCRIPT

A. REASON FOR USING SIMSCRIPT

Before discussing the suitability and practicability of SIMSCRIPT, it seems appropriate to explain why FORTRAN was not used to describe the revised model, since it had been used for the original model. The reason for not using FORTRAN was simply to determine if a general purpose language (i.e., FORTRAN) is less efficient than a simulation programming language when applied to an internal control system simulation model. Efficiency is discussed in terms of the suitability and practicability of SIMSCRIPT relative to FORTRAN. It should be emphasized that the "SIMSCRIPT" referred to in this thesis is SIMSCRIPT II.5, which is the proprietary version of SIMSCRIPT II marketed by Consolidated Analysis Centers, Inc.[7]

The decision to select SIMSCRIPT from all the simulation programming languages available was based on a review of the literature regarding simulation languages. One source stated that SIMSCRIPT is "the most comprehensive simulation language available".[8] This statement suggested that if SIMSCRIPT was not suitable to the problem, no simulation language would be

---

[7] SIMSCRIPT II.5 is a trademark of C.A.C.I., 12011 San Vicente Boulevard, Los Angeles, California, 90049.

[8] Fishman, G. S., Concepts and Methods in Discrete Event Digital Simulation, p. 70, Wiley, 1973.

suitable. Another source states that SIMSCRIPT "can do anything that can be done in FORTRAN".[9] This was also important, since it provided some assurance that an attempt to use SIMSCRIPT would not be futile. When the statements of these two sources were considered together, selection of SIMSCRIPT as an experimental language seemed to be a logical course of action.

An overview of the SIMSCRIPT language is provided by the preface of the book from which the author learned SIMSCRIPT. The preface states that the language can be considered to have five levels:

> Level 1: A simple teaching language designed to introduce programming concepts to non-programmers.
> Level 2: A language roughly comparable in power with FORTRAN, but departing greatly from it in specific features.
> Level 3: A language roughly comparable in power to ALGOL or PL/I, but again with many specific differences.
> Level 4: That part of SIMSCRIPT II that contains the entity-attribute-set features of SIMSCRIPT. These features have been updated and augmented to provide a more powerful list-processing capability. This level also contains a number of new data types and programming features.
> Level 5: The simulation-oriented part of SIMSCRIPT II, containing statements for time advance, event-processing, generation of statistical variates, and accumulation and analysis of simulation-generated data.[10]

---

[9] Emshoff, J. R. and Sisson, R. L., Design and Use of Computer Simulation Models, p. 140, Macmillan, 1970.

[10] Kiviat, P. J., Villanueva, R., and Markowitz, H. M., SIMSCRIPT II.5 Programming Language, p. v, Consolidated Analysis Centers Inc., 1973.

B. SUITABILITY OF SIMSCRIPT

Before discussing the suitability of SIMSCRIPT to the simulation of the internal control system of the hypothetical firm, it would be useful to define suitability. For the purpose of the following discussion, suitability is defined as the ease with which SIMSCRIPT can be applied to the problem of describing the revised simulation model.

1. Strengths of SIMSCRIPT

The primary strength of SIMSCRIPT has already been mentioned: it can do anything FORTRAN can do. Additional strengths of SIMSCRIPT lie in the fact that it can perform many of the routine operations required in a simulation program with greater ease than FORTRAN. When the revised simulation model was described in SIMSCRIPT, three examples of these routine operations were encountered.

The first example is the generation of random variates from a specified distribution. A normal distribution was specified in the QUANT1 routine in a single SIMSCRIPT statement which took the place of several FORTRAN statements. The normal distribution was only one of the eleven built-in random variate generators which SIMSCRIPT provides. Other statistical distribution functions available include the Erlang, exponential, Poisson, uniform, and Weibull distributions. The ready availability of all these and other distributions provide a programmer with many options while describing a simulation model.

The second example is the generation of pseudo-random numbers. In addition to being used in the QUANT1 routine to generate random variates, the random number generator was used in the ERROR routines. Each of the three ERROR routines in the revised simulation model had statements which generated pseudo-random numbers and used them in a Monte Carlo simulation of the error processes. The fact that SIMSCRIPT's built-in random number generator has been shown to produce good sampling properties means that the programmer can avoid having to perform tests of independence and uniformity.[11]

The third and final example of routine simulation operations easily performed is the calculation of statistics. Just one SIMSCRIPT statement calculated the mean and standard deviation of the error in the ending account balances. These two statistics represent only a portion of SIMSCRIPT's eleven built-in statistical computation routines.

2. Weakness of SIMSCRIPT

Because SIMSCRIPT is a more powerful language than FORTRAN, it has more features, and hence takes longer to learn. However, the author feels that two factors can reduce the significance of this weakness. First, if a person knows what he wants to do with SIMSCRIPT, he can skip over the features which he will not use. This is especially true if an internal control system is being simulated, since such a model would not utilize many of the features which are

[11] Fishman, op. cit., pp. 183-4.

more difficult to learn. A few examples of these features are formatted reports, the text data mode, the timing functions, and some of the more complex aspects of entities and attributes. The second factor is that if a person has already learned a general purpose programming language, he will find SIMSCRIPT very easy to learn. The author was very familiar with PL/I, and found SIMSCRIPT to be quite similar in many respects. Since the author benefited from both of these factors just mentioned, he spent less than four hours teaching himself the basics of SIMSCRIPT.

3. Assessment of Suitability

The strengths of SIMSCRIPT made it extremely suitable for simulating the revised model presented in this thesis. The one weakness of SIMSCRIPT seems trivial when compared to these strengths, especially in view of the two factors which mitigate the weakness. In addition, it should be noted that SIMSCRIPT has many capabilities which might be useful in a more complex simulation model of an internal control system. Perhaps the best example of these capabilities is the abundance of reliable built-in random variate generators. Furthermore, SIMSCRIPT can with little difficulty generate random variables based on any step function or linear function which the programmer describes. Thus it can be concluded that SIMSCRIPT is suitable for both the problem dealt with in this thesis and with more complex problems.

C. PRACTICABILITY OF SIMSCRIPT

For the purpose of the following discussion practicability will be defined as the feasibility of using SIMSCRIPT to solve the problem of describing the revised simulation model. This discussion assumes that SIMSCRIPT is suitable for solving the problem and deals with how efficiently (relative to FORTRAN) it solves the problem.

1. Strengths of SIMSCRIPT

There are three strengths in the practicability of SIMSCRIPT. First of all, SIMSCRIPT programs are much more readable than FORTRAN programs and thus contain more self-documentation. Second, the author found SIMSCRIPT to be easier to debug than FORTRAN. Third, the author felt that it was easier to express the simulation model in SIMSCRIPT than it would have been to express it in FORTRAN. The author felt this way primarily because the built-in functions provided by SIMSCRIPT reduced the amount of programming he was required to do. If FORTRAN routines had been available to generate random numbers, perform statistical computations, and generate random variates, SIMSCRIPT would have been only slightly easier to use than FORTRAN.

2. Weaknesses of SIMSCRIPT

The major weakness in the feasibility of SIMSCRIPT was the fact that the SIMSCRIPT program required significantly more computer time than a comparable FORTRAN program. For the compile step, the SIMSCRIPT program required 18.11 seconds while the FORTRAN program required only 10.51 seconds.

FORTRAN also had the advantage in the assemble and link steps (0.82 seconds and 1.03 seconds to SIMSCRIPT's 2.78 and 2.03 seconds, respectively). However, the worst comparison for SIMSCRIPT is execution efficiency during the go step, where it took about five times as long as FORTRAN (6 minutes and 57.22 seconds to 1 minute and 22.38 seconds). When total computer time was compared, SIMSCRIPT required about 4.6 times longer than FORTRAN.

There were several additional weaknesses in the practicability of SIMSCRIPT, but all were minor in comparison to that mentioned above. The first such weakness was the fact that a SIMSCRIPT compiler requires a relatively larger amount of core than a FORTRAN compiler. A second weakness was the fact that there are far fewer programmers with SIMSCRIPT experience than with FORTRAN experience. This makes it difficult to obtain programming assistance. A third weakness was the fact that there is much less documentation available regarding SIMSCRIPT than is available regarding FORTRAN. While none of these weaknesses are very significant by themselves, they do become significant when combined with the larger amount of computer time which SIMSCRIPT requires.

3. Assessment of Practicability

In order to assess the practicability of using SIMSCRIPT to simulate the internal control system of the hypothetical firm, it is necessary to compare the benefits derived from the strengths with the costs incurred due to

the weaknesses. The author's conclusion is that the costs outweighed the benefits. This means that it would have been more practicable to use FORTRAN for this particular simulation than it was to use SIMSCRIPT. However, the author feels that if a larger, more complicated model were to be built, SIMSCRIPT would be more practicable for that model than it was for this model.

D. FINAL ASSESSMENT OF SIMSCRIPT

This final assessment will consider the strengths and weaknesses of SIMSCRIPT with respect to both suitability and practicability. Since it has already been concluded that SIMSCRIPT was more suitable but less practicable than FORTRAN, some net assessment seems in order. Because of the particular circumstances under which this computer simulation model was built, the net assessment of the author was that SIMSCRIPT was preferable to FORTRAN.

Circumstances peculiar to this academic exercise were that the author considered his time to be a relatively scarce (and therefore, costly) resource, while computer time was a relatively abundant, low-cost resource. This illustrates that the tradeoff between personnel costs and computer costs should be considered when making a decision regarding whether to use SIMSCRIPT or FORTRAN. It should be emphasized that in a real-world situation it would have been less costly to build the model in FORTRAN than in SIMSCRIPT when all costs were considered. This statement reflects the author's

opinion that the use of SIMSCRIPT would not reduce programming costs enough to offset the cost of the additional computer time required.

V. SUMMARY AND CONCLUSION

A. METHODOLOGY FOR SIMULATING INTERNAL CONTROL

1. Success of Revised Methodology

The major conclusion of this thesis is that the revised simulation methodology it presents can provide a more efficient method of objectively evaluating internal control than the original methodology used by Burns. The greater efficiency of the revised methodology is due to the fact that the error routines are more general in nature than the original simulation program. Greater generality increases the probability that the same error routines can be used to define other accounting and internal control systems.

The fact that the SIMSCRIPT program used to describe the revised model required more computer time than a similar FORTRAN program should not be used to argue that the revised methodology is less efficient than the original methodology. This is true because the revised methodology could almost as easily have used FORTRAN. As was previously stated, the use of SIMSCRIPT was completely independent of the use of the revised methodology.

In addition, the argument that using FORTRAN to describe the revised model would be more costly than using FORTRAN to describe the original model does not refute the efficiency of the revised methodology. This is true because the revised methodology is intended to reduce the cost of describing internal control systems in addition to the one

dealt with in this thesis. The author admits that a high set-up cost is incurred when programming the general purpose error routines. However, once the error routines are written, the cost which is incurred is that of arranging the routines in the proper order and inserting the necessary arguments into them. Consequently, the cost of preparing a simulation model of an internal control system using the revised methodology will be much less costly and much more efficient once the general error routines have been written.

A concluding comment seems to be in order regarding the significance of the output of an internal control system simulation model prepared using the revised methodology. This output would be represented by the mean and standard deviation of the error present in the ending balance of each account contained in the simulation model. These two statistics provide an objective measure which can be used to determine the adequacy of internal control in the system being simulated. An assessment of adequacy can be made only after the materiality of the error is determined when considering the reported account balances of the real firm.

2. Extension of Revised Methodology

Upon completion of the research performed as a basis for this thesis, the author recognized one direction in which this revised methodology could be extended. This extension related to the fact that the error processes in the hypothetical firm were assumed to have discrete probability distributions. Since real world error processes are likely

to be continuous, it might well be true that error processes could be more easily represented by continuous probability density functions. If this were the case, one of the arguments for each general error routine would be the probability density function of the error's magnitude. Such an argument might possibly reduce the number of general error routines that would be needed. At any rate, the author suggests that further research should be conducted in the area of developing error routines which are more general.

B. FEASIBILITY OF USING SIMSCRIPT

The use of SIMSCRIPT to describe the revised simulation model provided the author with an opportunity to assess the feasibility of using this language to describe simulation models of internal control systems using the revised methodology. The conclusion which the author reached was that SIMSCRIPT was more suitable than FORTRAN, but that SIMSCRIPT required more computer time than FORTRAN. Thus the benefit of requiring less programming time was offset against the requiring of more computer time. The author feels that unless more complex simulation models are constructed, SIMSCRIPT would not need to be used. It is possible that some other language (such as PL/I) might provide a less expensive alternative than SIMSCRIPT. This is another area where future research should be performed.

C. ISSUES TO BE RESOLVED

One of the purposes of this thesis was to identify the basic requirements for a language which could be used specifically for the simulation of internal control systems. Upon completion of this research, the author concluded that several issues would have to be resolved before such basic requirements could be identified. Unfortunately, resolution of these issues required information the author could not obtain, so it is possible only to state the issues which must be resolved.

1. Optimal Size System to Simulate

One of the first issues which must be resolved deals with the problem of deciding the optimal size internal control system to simulate. A good measure of size would be the number of accounts contained in the system. For example, the inventory accounting system dealt with in this thesis was basically concerned with three inventory accounts. Two additional accounts collected miscellaneous debits and credits. An alternative system could have included all accounts which were debited or credited in the miscellaneous accounts. One such account would be accounts payable.

However, it might be more appropriate to simulate the entire internal control system related to accounts payable separately from the inventory accounting internal control system. Thus it can be seen that a decision must be made regarding whether it is better to simulate each internal

control subsystem separately or to simulate the entire internal control system of a firm in one model.

Regardless of the decision which is reached, the author has two suggestions which would simplify the simulation process. First of all, the person designing the simulation model should not concern himself with accounts whose transactions are so few in number that subjective assessment of the adequacy of internal control is feasible. Examples of such accounts would be plant, property, and equipment, owner's equity accounts, and long-term debt. Second, preparation of a comprehensive chart of accounts would allow the auditor to avoid having to set up each account involved in the simulation. In addition, verifying a zero balance in the accounts which were not involved would provide a control to insure that all errors were recorded in the proper accounts.

2. Conceptualization of Errors

Another issue which must be resolved deals with the manner in which auditors conceptualize errors. An internal control simulation language should be constructed in a manner which would allow auditors to easily express error processes in the manner in which they conceptualize them. In the model described in this thesis, all errors were conceptualized as either count (or quantity) errors or rate (or price) errors. It might be true that auditors conceive the monetary value of errors to be distributed according to the normal distribution or some other statistical distribution.

On the other hand, they might consider the distribution of monetary errors detected during compliance tests to represent a likely error probability density function. Another alternative might be that auditors think in terms that could be expressed in the form of the beta distribution used in PERT networks. If auditors construct optimistic, most likely, and pessimistic error values, these estimates could be used to determine the expected distribution of error.

Once it has been determined how auditors conceptualize errors, routines could be written to simulate these error processes. It should be pointed out that writing too few routines might be more dangerous than writing too many. Too few routines would cause auditors to discount the usefulness of simulation models, since they would have to fit the real system into the model rather than fitting the model to the real system. Too many routines would simply provide an overly powerful (and thus, more expensive) language than the auditor needs. Presumably there are a limited number of ways in which errors are conceptualized by auditors, so any routines which would not be used would simply add to the expense of performing simulations.

3. Conceptualization of Controls

In a similar fashion, the manner in which controls are conceptualized by auditors must be determined. Since the author found programming the internal controls to be one of the most difficult aspects of the simulation, he feels this

area needs more attention. In the model presented in this thesis, the control itself could interject error into the accounting system because of the manner in which it attempted to correct previous errors. Thus it should be determined whether auditors normally conceptualize controls as capable of interjecting errors or whether they consider controls to simply set the possible error equal to zero. A recent article by Barry Cushing sheds some light on this subject by presenting one method of conceptualizing the interaction of controls and errors.[12]

This issue is related to the conceptualization of errors because it might be possible to reflect the effect of controls in the error processes. This would be possible by simply having the auditor express errors in terms of the error distribution which would be expected after the control has been encountered. Unfortunately, it may not be possible to do this in many of the more complex internal control systems.

Finally, it should be determined what criteria are normally used to decide whether or not a given control will be encountered. For example, the original model assumed that the number of units by which production orders were overstated determined whether or not an error was detected. In other situations dollar values rather than unit quantities

---

[12] Cushing, B. E., "A Mathematical Approach to the Analysis and Design of Internal Control Systems," The Accounting Review, v. XLIX, pp. 24-41, January 1974.

might be appropriate criteria. Whatever other criteria are determined to be commonly found in internal control systems should be provided for in an internal control simulation language.

4. Structure of Input Data

The final issue which needs to be resolved deals with how the input data should be structured in an internal control simulation language. In the model contained in this thesis some data was specified in the main program (i.e., number of accounts, number of iterations, activity ceilings, etc.), while some data was contained in the arguments used when calling error routines. The author found that there was a tradeoff between the number of arguments used in an error routine and the number of error routines which were necessary to describe all possible error processes. At the limit, this meant that one error routine could have been written to describe the entire system, but every variable in the model would have to be an argument of this routine.

Since the author felt that the likelihood of an error in input data increased with the number of arguments, he considered the use of only one error routine to be impractical. However, a viable alternative would be for the simulation program to be designed so that only input data need be prepared by the person describing the system. The simulation program would then begin with a routine that would read in the input data and check it to make certain that each variable was given a value. If such a routine were included in

an internal control simulation language it could also generate error diagnostics regarding invalid or illogical input data. The language might also include an option which would allow the use of empirical data from the firm rather than approximating such data by using random variate generators.

D. FINAL REMARKS

The author feels that once the issues stated above have been resolved, it will be possible to determine the requirements for a language to be used for the simulation of internal control systems. Simulation models prepared with this language would allow an auditor to supplement the traditional subjective methods of assessing the adequacy of internal control with more objective methods. Since the superiority of the simulation methodology over conventional auditing procedures has already been clearly demonstrated in Burns' dissertation, the author feels that the time has come to develop an internal control system simulation language.

APPENDIX A

OPERATIONAL FLOW CHART

OF REVISED SIMULATION MODEL

Establish routines and global variables.

↓

Specify the number of accounts to be simulated and the number of iterations to be performed.

↓

Dimension computer memory for all subscripted variables.

↓

At the beginning of each iteration initialize account error balances to zero and establish the seed value for generating raw material shipment and production order quantities.

↓

For each raw material, set a counter equal to zero and specify the ceiling which will control the number of raw material shipments to be generated.

↓

Call routine QUANT1 to generate a value for the number of units contained in a shipment.

↓

Call routine ERROR1 to generate a count error and price error and determine if these errors shall occur. If an error does occur, calculate its monetary value.

↓

Call routine ENTRY1 to record the value of the error in the proper accounts.

↓

Increment the counter by the number of units actually contained in the shipment. Test to determine if the ceiling for raw material shipments has been exceeded. If the ceiling has been exceeded, proceed to the next raw material. If the last raw material has been simulated, proceed to the simulation of the production process.

↓

For each product, set a counter equal to zero and specify the ceilings which will control the number of units to be produced and the number of units to be transferred to finished goods.

↓

Call routine QUANT1 to generate a value for the number of units to be indicated on a production order in Department I.

↓

Call routine ERROR1 to generate a count error and price error in costing-out raw material and determine if these errors shall occur. If an error does occur, calculate its monetary value.

↓

Call routine ENTRY1 to record the value of the error in the proper accounts.

↓

Call routine ERROR2 to generate a rate error in Department I burden and determine if this error shall occur. If the error does occur, calculate its monetary value.

↓

Call routine ERROR3 to generate a rate error in Department I labor and determine if this error shall occur. If the error does occur, calculate its monetary value and add it to the monetary value of the error calculated in ERROR2 above.

↓

Call routine ENTRY2 to record the value of the error in the proper accounts.

↓

Call routine ERROR1 to generate a count error in the production count for Department II and determine if this error shall occur. If the error does occur, calculate its monetary value.

↓

Call routine ERROR2 to generate a rate error in Department II burden and determine if this error shall occur. If the error does occur, calculate its monetary value.

↓

Call routine ERROR3 to generate a rate error in Department II labor and determine if this error shall occur. If the error does occur, calculate its monetary value and add it to the monetary value of the error calculated in ERROR2 above.

↓

Call routine ENTRY3 to record the value of the error in the proper accounts.

↓

Determine if the ceiling for number of units to be transferred to finished goods has been exceeded. If this ceiling has been exceeded, do not transfer these units to finished goods, but do continue simulating the production process.

↓

If the ceiling for the number of units to be transferred to finished goods has not been exceeded, simulate the control performed by the weigh-count operator. This control will detect an overstatement of the count on a production order if its magnitude is greater than twenty units.

↓

If the control detects an overstatement in the production count, it corrects the count on the production order and sends the production order to the accounting department. If no overstatement is detected, no correction is made when the production order is forwarded.

↓

Simulate the operations performed in the accounting department. This is done by calling the routine ERROR1 to generate a pricing error and determine if this error is to occur. If the error does occur, calculate its monetary value.

↓

Call routine ENTRY3 to record the value of the error in the proper accounts.

↓

Simulate the corrections made by the accounting department if the control detected an overstatement. This is done by calculating the adjustments necessary to correct the errors which were assumed to be present in the raw materials and work-in-process accounts. The monetary value of these errors is calculated and routine ENTRY3 is called to record the value of the error in the proper accounts.

↓

Increment the counter by the number of units stated on the production order. Test to determine if the ceiling for number of units to be produced has been exceeded. If the ceiling has been exceeded, proceed to the next product. If the last product has been simulated, proceed to the section of the program which prints the output.

↓

Print out the ending account balance, which contains the monetary value of the error in the ending account balance. After this has been done for each account, store the value in an array for later processing.

↓

Determine if the required number of iterations have been performed. If they have not all been performed, proceed to the beginning of the program.

↓

If the required number of iterations have been performed, calculate the mean and standard deviation of the error in the account balances which were stored earlier. Upon calculation of these statistics, print them out.

```
//SIMRUN$2 JOB (2471,0234,6418),'C.M.PON',TIME=(14,30)
// EXEC SIM25CLG,REGION.GO=100K
//SIM.SYSIN DD *


PREAMBLE
NORMALLY MODE IS REAL
DEFINE INITIAL.QUANTITY AS AN INTEGER VARIABLE
DEFINE ERROR AS A REAL VARIABLE
DEFINE ACCOUNT AS A REAL 1-DIMENSIONAL ARRAY
DEFINE ACCT.TITLE AS AN ALPHA 1-DIMENSIONAL ARRAY
DEFINE SAMPLE AS A REAL 2-DIMENSIONAL ARRAY
DEFINE NO.OF.ITERATIONS AS AN INTEGER VARIABLE
DEFINE ERROR1 AS A ROUTINE WITH 5 ARGUMENTS
DEFINE QUANT1 AS A ROUTINE WITH 3 ARGUMENTS
DEFINE ENTRY1 AS A ROUTINE WITH 2 ARGUMENTS
DEFINE COUNTER AND QUOTA AS INTEGER VARIABLES
DEFINE NUMBER.OF.ACCOUNTS AS AN INTEGER VARIABLE
DEFINE ENTRY2 AS A ROUTINE WITH 2 ARGUMENTS
DEFINE ENTRY3 AS A ROUTINE WITH 2 ARGUMENTS
DEFINE ERROR2 AS A ROUTINE WITH 6 ARGUMENTS
DEFINE ERROR3 AS A ROUTINE WITH 5 ARGUMENTS
DEFINE COUNT.ERROR AS AN INTEGER VARIABLE
DEFINE HOURS AND RIGHT.HOURS AS REAL VARIABLES
DEFINE TOTAL.COUNT.ERROR AS AN INTEGER VARIABLE
DEFINE QUOTA1 AND QUOTA2 AS INTEGER VARIABLES
LAST COLUMN IS 60
END


MAIN
LET NUMBER.OF.ACCOUNTS = 6
LET NO.OF.ITERATIONS = 100
RESERVE SAMPLE AS NUMBER.OF.ACCOUNTS BY NO.OF.ITERATIONS
RESERVE ACCOUNT AND ACCT.TITLE AS NUMBER.OF.ACCOUNTS
FOR J = 1 TO NO.OF.ITERATIONS DO                        ''BEGIN MAIN''
FOR I = 1 TO NUMBER.OF.ACCOUNTS DO LET ACCOUNT(I) = 0.0 LOOP
LET SEED.V(2) = 683743814
LET ACCT.TITLE(1) = "MISC. DEBIT"
LET ACCT.TITLE(2) = "RAW MATERIAL"
LET ACCT.TITLE(3) = "WORK IN PROCESS"
LET ACCT.TITLE(4) = "FINISHED GOODS"
LET ACCT.TITLE(5) = "MISC. CREDIT"
LET ACCT.TITLE(6) = "COMBINED INVENTORY"
LET COUNTER = 0                                         ''RAW MATERIAL 1''
LET QUOTA = 40000
'BEGIN1'
CALL QUANT1(200.0,25.0,2)
CALL ERROR1(.25,-.10,.10,13.50,6.50)
CALL ENTRY1(2,5)
LET COUNTER = COUNTER + INITIAL.QUANTITY
IF COUNTER < QUOTA GO TO BEGIN1
ELSE
LET COUNTER = 0                                         ''RAW MATERIAL 2''
LET QUOTA = 34000
'BEGIN2'
CALL QUANT1(180.0,30.0,2)
CALL ERROR1(.25,-.10,.10,16.70,8.00)
CALL ENTRY1(2,5)
LET COUNTER = COUNTER + INITIAL.QUANTITY
IF COUNTER < QUOTA GO TO BEGIN2
ELSE
LET COUNTER = 0                                         ''RAW MATERIAL 3''
LET QUOTA = 34000
'BEGIN3'
CALL QUANT1(200.0,25.0,2)
CALL ERROR1(.25,-.10,.10,6.50,13.50)
CALL ENTRY1(2,5)
LET COUNTER = COUNTER + INITIAL.QUANTITY
IF COUNTER < QUOTA GO TO BEGIN3
ELSE
LET COUNTER = 0                                         ''RAW MATERIAL 4''
```

```
LET QUOTA = 32000
'BEGIN4'
CALL QUANT1(180.0,30.0,2)
CALL ERROR1(.25,-.10,.10,8.00,16.70)
CALL ENTRY1(2,5)
LET COUNTER = COUNTER + INITIAL.QUANTITY
IF COUNTER < QUOTA GO TO BEGIN4
ELSE
LET COUNTER = 0                                          ''PRODUCT 1''
LET QUOTA1 = 33600
LET QUOTA2 = 30280
'BEGIN.P1'
CALL QUANT1(150.0,35.0,2)                    ''DEPT. 1 PRODUCTION''
CALL ERROR1(.15,.10,.10,13.50,6.50)          ''DEPT. 1 DIR. MATL.''
CALL ENTRY1(3,2)
CALL ERROR2(.08,.06,.04,.08,12.85,11.40)  ''DEPT. 1 BURDEN''
CALL ERROR3(.10,6.00,5.60,.50,6.20)       ''DEPT. 1 DIR. LABOR''
CALL ENTRY2(3,5)
CALL ERROR1(.08,.05,0.0,0.0,0.0)             ''DEPT. 2 PRODUCTION''
LET INITIAL.QUANTITY = INITIAL.QUANTITY - TOTAL.COUNT.ERROR
LET TOTAL.COUNT.ERROR = TOTAL.COUNT.ERROR + COUNT.ERROR
LET COUNT.ERROR = TOTAL.COUNT.ERROR
CALL ERROR2(.08,.04,.04,.08,51.55,44.05)  ''DEPT. 2 BURDEN''
CALL ERROR3(.10,5.40,6.20,.50,5.60)       ''DEPT. 2 DIR. LABOR''
CALL ENTRY3(3,5)
IF COUNTER >= QUOTA2 AND COUNTER < QUOTA1
  GO TO TEST.P1
ELSE
IF TOTAL.COUNT.ERROR > 20      ''THE CHECK DETECTS AN ERROR''
LET COUNT.ERROR = 0
CALL ERROR1(-1.0,0.0,.08,16.929,9.19)
CALL ENTRY3(4,3)
LET ERROR = TOTAL.COUNT.ERROR * 3.429
CALL ENTRY3(1,3)
LET ERROR = TOTAL.COUNT.ERROR * 13.50
CALL ENTRY3(2,3)
GO TO TEST.P1                              ''END OF ERROR CORRECTION''
ELSE
CALL ERROR1(-1.0,0.0,.08,16.929,9.19)
CALL ENTRY3(4,3)
'TEST.P1'
LET COUNTER = COUNTER + INITIAL.QUANTITY
IF COUNTER < QUOTA1 GO TO BEGIN.P1
ELSE
LET COUNTER = 0                                          ''PRODUCT 2''
LET QUOTA1 = 33000
LET QUOTA2 = 30964
'BEGIN.P2'
CALL QUANT1(150.0,35.0,2)                    ''DEPT. 1 PRODUCTION''
CALL ERROR1(.15,.10,.10,16.70,8.00)          ''DEPT. 1 DIR. MATL.''
CALL ENTRY1(3,2)
CALL ERROR2(.08,.09,.06,.08,12.85,11.40)  ''DEPT. 1 BURDEN''
CALL ERROR3(.10,6.00,5.60,.50,6.20)       ''DEPT. 1 DIR.LABOR''
CALL ENTRY2(3,5)
CALL ERROR1(.08,.05,0.0,0.0,0.0)             ''DEPT. 2 PRODUCTION''
LET INITIAL.QUANTITY = INITIAL.QUANTITY - TOTAL.COUNT.ERROR
LET TOTAL.COUNT.ERROR = TOTAL.COUNT.ERROR + COUNT.ERROR
LET COUNT.ERROR = TOTAL.COUNT.ERROR
CALL ERROR2(.08,.09,.06,.08,51.55,44.05)  ''DEPT. 2 BURDEN''
CALL ERROR3(.10,5.40,6.20,.50,5.60)       ''DEPT. 2 DIR. LABOR''
CALL ENTRY3(3,5)
IF COUNTER >= QUOTA2 AND COUNTER < QUOTA1
  GO TO TEST.P2
ELSE
IF TOTAL.COUNT.ERROR > 20      ''THE CHECK DETECTS AN ERROR''
LET COUNT.ERROR = 0
CALL ERROR1(-1.0,0.0,.08,21.8435,12.5315)
CALL ENTRY3(4,3)
LET ERROR = TOTAL.COUNT.ERROR * 5.1435
CALL ENTRY3(1,3)
LET ERROR = TOTAL.COUNT.ERROR * 16.70
CALL ENTRY3(2,3)
```

```
GO TO TEST.P2                          ''END OF ERROR CORRECTION''
ELSE
CALL ERROR1(-1.0,0.0,.08,21.8435,12.5315)
CALL ENTRY3(4,3)
'TEST.P2'
LET COUNTER = COUNTER + INITIAL.QUANTITY
IF COUNTER < QUOTA1 GO TO BEGIN.P2
ELSE
LET COUNTER = 0                                     ''PRODUCT 3''
LET QUOTA1 = 33000
LET QUOTA2 = 29777
'BEGIN.P3'
CALL QUANT1(150.0,35.0,2)               ''DEPT. 1 PRODUCTION''
CALL ERROR1(.15,.10,.10,6.50,13.50)     ''DEPT. 1 DIR. MATL.''
CALL ENTRY1(3,2)
CALL ERROR2(.08,.04,.06,.08,11.40,12.85)  ''DEPT. 1 BURDEN''
CALL ERROR3(.10,6.00,5.60,.50,6.20)     ''DEPT. 1 DIR. LABOR''
CALL ENTRY2(3,5)
CALL ERROR1(.08,.05,0.0,0.0,0.0)        ''DEPT. 2 PRODUCTION''
LET INITIAL.QUANTITY = INITIAL.QUANTITY - TOTAL.COUNT.ERROR
LET TOTAL.COUNT.ERROR = TOTAL.COUNT.ERROR + COUNT.ERROR
LET COUNT.ERROR = TOTAL.COUNT.ERROR
CALL ERROR2(.08,.04,.06,.08,44.05,51.55)  ''DEPT. 2 BURDEN''
CALL ERROR3(.10,5.40,6.20,.50,5.60)     ''DEPT. 2 DIR. LABOR''
CALL ENTRY3(3,5)
IF COUNTER >= QUOTA2 AND COUNTER < QUOTA1
  GO TO TEST.P3
ELSE
IF TOTAL.COUNT.ERROR > 20      ''THE CHECK DETECTS AN ERROR''
LET COUNT.ERROR = 0
CALL ERROR1(-1.0,0.0,.08,9.19,16.929)
CALL ENTRY3(4,3)
LET ERROR = TOTAL.COUNT.ERROR * 2.69
CALL ENTRY3(1,3)
LET ERROR = TOTAL.COUNT.ERROR * 6.50
CALL ENTRY3(2,3)
GO TO TEST.P3                          ''END OF ERROR CORRECTION''
ELSE
CALL ERROR1(-1.0,0.0,.08,9.19,16.929)
CALL ENTRY3(4,3)
'TEST.P3'
LET COUNTER = COUNTER + INITIAL.QUANTITY
IF COUNTER < QUOTA1 GO TO BEGIN.P3
ELSE
LET COUNTER = 0                                     ''PRODUCT 4''
LET QUOTA1 = 28900
LET QUOTA2 = 21719
'BEGIN.P4'
CALL QUANT1(150.0,35.0,2)               ''DEPT. 1 PRODUCTION''
CALL ERROR1(.15,.10,.10,8.00,16.70)     ''DEPT. 1 DIR. MATL.''
CALL ENTRY1(3,2)
CALL ERROR2(.08,.06,.09,.08,11.40,12.85)  ''DEPT. 1 BURDEN''
CALL ERROR3(.10,6.00,5.60,.50,6.20)     ''DEPT. 1 DIR. LABOR''
CALL ENTRY2(3,5)
CALL ERROR1(.08,.05,0.0,0.0,0.0)        ''DEPT. 2 PRODUCTION''
LET INITIAL.QUANTITY = INITIAL.QUANTITY - TOTAL.COUNT.ERROR
LET TOTAL.COUNT.ERROR = TOTAL.COUNT.ERROR + COUNT.ERROR
LET COUNT.ERROR = TOTAL.COUNT.ERROR
CALL ERROR2(.08,.06,.09,.08,44.05,51.55)  ''DEPT. 2 BURDEN''
CALL ERROR3(.10,5.40,6.20,.50,5.60)     ''DEPT. 2 DIR. LABOR''
CALL ENTRY3(3,5)
IF COUNTER >= QUOTA2 AND COUNTER < QUOTA1
  GO TO TEST.P4
ELSE
IF TOTAL.COUNT.ERROR > 20      ''THE CHECK DETECTS AN ERROR''
LET COUNT.ERROR = 0
CALL ERROR1(-1.0,0.0,.08,12.5315,21.8435)
CALL ENTRY3(4,3)
LET ERROR = TOTAL.COUNT.ERROR * 4.5315
CALL ENTRY3(1,3)
LET ERROR = TOTAL.COUNT.ERROR * 8.00
CALL ENTRY3(2,3)
```

```
GO TO TEST.P4                              ''END OF ERROR CORRECTION''
ELSE
CALL ERROR1(-1.0,0.0,.06,12.5315,21.8435)
CALL ENTRY3(4,3)
'TEST.P4'
LET COUNTER = COUNTER + INITIAL.QUANTITY
IF COUNTER < QUOTA1 GO TO BEGIN.P4
ELSE
'OUTPUT'          ''USER NEED NOT SPECIFY THIS''
LET ACCOUNT(6) = ACCOUNT(2) + ACCOUNT(3) + ACCOUNT(4)
FOR I = 1 TO NUMBER.OF.ACCOUNTS DO
PRINT 1 LINE WITH ACCT.TITLE(I) AND ACCOUNT(I) THUS
****************   *******.**
  LOOP
  SKIP 1 OUTPUT LINE
FOR I = 1 TO NUMBER.OF.ACCOUNTS DO
  LET SAMPLE(I,J) = ACCOUNT(I)
  LOOP
LOOP                                                       ''END MAIN''
FOR I = 1 TO NUMBER.OF.ACCOUNTS DO
  FOR J = 1 TO NO.OF.ITERATIONS DO
  COMPUTE XBAR AS THE MEAN AND SIGMA AS THE STD.DEV OF  .
  SAMPLE(I,J)
  LOOP
PRINT 1 LINE WITH ACCT.TITLE(I), XBAR, AND SIGMA THUS
  ****************   *******.**   *******.**
  LOOP
END


ROUTINE QUANT1(MEAN,STD.DEV,STREAM)
LET INITIAL.QUANTITY = NORMAL.F(MEAN,STD.DEV,STREAM)
LET TOTAL.COUNT.ERROR = 0
END


ROUTINE ERROR1(PROB.OF.COUNT.ERROR,COUNT.ERROR.PERCENT,
  PROB.OF.PRICE.ERROR,CORRECT.PRICE,WRONG.PRICE)
NORMALLY MODE IS REAL
IF PROB.OF.COUNT.ERROR < 0.0
  LET R1 = PROB.OF.COUNT.ERROR - 1.0
  GO TO NEXT1              ''COUNT ERROR ALREADY HAS A VALUE''
ELSE
LET R1 = RANDOM.F(3)
IF R1 < PROB.OF.COUNT.ERROR
  LET CCUNT.ERROR = INITIAL.QUANTITY * COUNT.ERROR.PERCENT
  GO TO NEXT1
ELSE LET COUNT.ERROR = 0
'NEXT1'
LET R2 = RANDOM.F(4)
IF R2 < PROB.OF.PRICE.ERROR
  LET PRICE.ERROR = WRONG.PRICE - CORRECT.PRICE
  GO TO NEXT2
ELSE LET PRICE.ERROR = 0.0
'NEXT2'
IF R1 < PROB.OF.COUNT.ERROR OR R2 < PROB.OF.PRICE.ERROR
  LET ERROR = (COUNT.ERROR * CORRECT.PRICE)
  + (PRICE.ERROR * (INITIAL.QUANTITY + COUNT.ERROR))
  GO TO 'NEXT3'
ELSE LET ERROR = 0.0
'NEXT3'
END


ROUTINE ERROR2(PROB.HRS.ERROR,RIGHT.DIR.LABOR.HRS,
  WRONG.DIR.LABOR.HRS,PROB.RATE.ERROR,CORRECT.RATE,
  WRONG.RATE)
LET R2 = RANDOM.F(5)
IF R2 < PROB.HRS.ERROR
  LET HOURS = WRONG.DIR.LABOR.HRS
  GO TO NEXT1
ELSE LET HOURS = RIGHT.DIR.LABOR.HRS
```

```
'NEXT1'
LET R3 = RANDOM.F(6)
IF R3 < PROB.RATE.ERROR
  LET RATE = WRONG.RATE
  GO TO NEXT2
ELSE LET RATE = CORRECT.RATE
'NEXT2'
LET ERROR = ERROR + (COUNT.ERROR * HOURS * RATE)
LET RATE.ERROR = ((INITIAL.QUANTITY + COUNT.ERROR)
  * ((HOURS * RATE) - (RIGHT.DIR.LABOR.HRS * CORRECT.RATE)))
LET ERROR = ERROR + RATE.ERROR
LET RIGHT.HOURS = RIGHT.DIR.LABOR.HRS
END

ROUTINE ERROR3(PROB.OF.RATE.ERROR,WRONG.RATE.1,WRONG.RATE.2,
  PROB.OF.WRONG.RATE.1,CORRECT.RATE)
NORMALLY MODE IS REAL
LET R3 = RANDOM.F(7)
IF R3 < PROB.OF.RATE.ERROR
  LET RATE = WRONG.RATE.2
  GO TO TEST1
ELSE LET RATE = CORRECT.RATE
GO TO FINISH
'TEST1'
LET R4 = RANDOM.F(8)
IF R4 < PROB.OF.WRONG.RATE.1
  LET RATE = WRONG.RATE.1
ELSE
'FINISH'
LET ERROR = ERROR + (COUNT.ERROR * HOURS * RATE)
LET RATE.ERROR = (INITIAL.QUANTITY + COUNT.ERROR)
  * ((HOURS * RATE) - (RIGHT.HOURS * CORRECT.RATE))
LET ERROR = ERROR + RATE.ERROR
END

ROUTINE ENTRY1(DEBIT,CREDIT)
NORMALLY MODE IS INTEGER
LET ACCOUNT(DEBIT) = ACCOUNT(DEBIT) + ERROR
LET ACCOUNT(CREDIT) = ACCOUNT(CREDIT) - ERROR
LET ERROR = 0.0
LET TOTAL.COUNT.ERROR = 0
END

ROUTINE ENTRY2(DEBIT,CREDIT)
NORMALLY MODE IS INTEGER.
LET ACCOUNT(DEBIT) = ACCOUNT(DEBIT) + ERROR
LET ACCOUNT(CREDIT) = ACCOUNT(CREDIT) - ERROR
LET INITIAL.QUANTITY = INITIAL.QUANTITY + COUNT.ERROR
LET TOTAL.COUNT.ERROR = TOTAL.COUNT.ERROR + COUNT.ERROR
LET ERROR = 0.0
END

ROUTINE ENTRY3(DEBIT,CREDIT)
NORMALLY MODE IS INTEGER
LET ACCOUNT(DEBIT) = ACCOUNT(DEBIT) + ERROR
LET ACCOUNT(CREDIT) = ACCOUNT(CREDIT) - ERROR
LET ERROR = 0.0
END
```

LIST OF REFERENCES

1. Burns, D. C., Audit Evidence Evaluation Using Computer Simulation with Special Emphasis on Ascertaining the Reliability of Accounting Data, Doctoral Dissertation, Indiana University, Bloomington, Graduate School of Business, 1971.

2. Committee on Auditing Procedure, Statement on Auditing Standards No. 1, American Institute of Certified Public Accountants, Inc., 1973.

3. Cushing, B. E., "A Mathematical Approach to the Analysis and Design of Internal Control Systems," The Accounting Review, v. XLIX, January 1974.

4. Emshoff, J. R. and Sisson, R. L., Design and Use of Computer Simulation Models, Macmillan, 1970.

5. Fishman, G. S., Concepts and Methods in Discrete Event Digital Simulation, Wiley, 1973.

6. Kiviat, P. J., Villanueva, R., and Markowitz, H. M., SIMSCRIPT II.5 Programming Language, p. v, Consolidated Analysis Centers, Inc., 1973.

7. Naval Postgraduate School Report 55Bu73111A, A Computer Simulation Case for the Auditing Classroom, by D. C. Burns, November 1973.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 55 Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
4.	Asst. Professor David C. Burns, Code 55Bu Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
5.	ENS Gerald G. Brown, USNR, Code 55Zr Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
6.	LTJG Carl Maynard Pon, USNR 1245 Leahy Road Monterey, California 93940	1

153120

Thesis
P7345 Pon
c.1 A methodology for the simulation of internal control systems.